(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 525 394 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.08.2019 Bulletin 2019/33

(51) Int Cl.:
H04L 9/32 *(2006.01)* ***G06Q 20/06*** *(2012.01)*

(21) Application number: **18155717.4**

(22) Date of filing: **08.02.2018**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA MD TN

(71) Applicant: **Sony Corporation**
Tokyo 108-0075 (JP)

(72) Inventors:
- **CRONIE, Harm**
  1015 Lausanne (CH)
- **NOLAN, Julian**
  1015 Lausanne (CH)
- **KOIKE, Makoto**
  Tokyo 108-0075 (JP)

(74) Representative: **MFG Patentanwälte**
Meyer-Wildhagen Meggle-Freund
Gerhard PartG mbB
Amalienstraße 62
80799 München (DE)

(54) ELECTRONIC DEVICES, SYSTEMS AND METHODS

(57) A method comprising determining if two separated distributed ledgers share a common history.

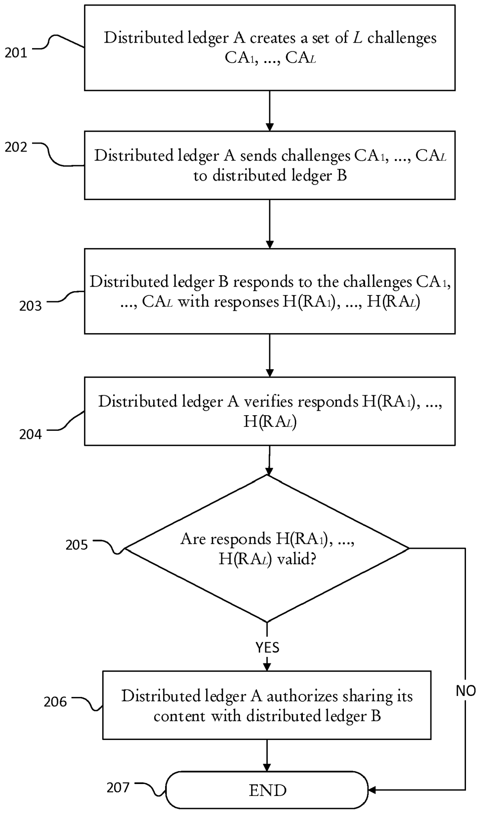
201
Distributed ledger A creates a set of *L* challenges $CA_1, ..., CA_L$
202
Distributed ledger A sends challenges $CA_1, ..., CA_L$ to distributed ledger B
203
Distributed ledger B responds to the challenges $CA_1, ..., CA_L$ with responses $H(RA_1), ..., H(RA_L)$
204
Distributed ledger A verifies responds $H(RA_1), ..., H(RA_L)$
205
Are responds $H(RA_1), ..., H(RA_L)$ valid?
YES
NO
206
Distributed ledger A authorizes sharing its content with distributed ledger B
207
END

Fig. 2

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

[0001] The present disclosure generally pertains to the field of electronic data storage, in particular to the storage of transactions in a distributed ledger such as a blockchain.

TECHNICAL BACKGROUND

[0002] A distributed ledger may for example be a distributed database, for example a distributed database that maintains a continuously growing list of data records secured from tampering and revision such as a blockchain. A blockchain consists of blocks that hold timestamped batches of valid transactions. In the following, the term transaction generally refers to a data entity that is stored as a record on the distributed ledger. A transaction may for example reflect a money transfer, a smart contract, an asset, or the like.

[0003] Blockchain technology can for example be used to track the history of money transactions (e.g. bitcoins), or it may be used to track or manage individual devices, by recording a ledger of data exchanges between the devices. Tracking or managing devices is also known under the term "Internet of Things" (IoT).

[0004] A large group of IoT devices may maintain a distributed ledger, e.g. a blockchain to record transactions (e.g. execute smart contracts). In such case, the nodes accessing the distributed ledger are devices. It may happen that a subgroup of nodes gets disconnected for a substantial amount of time from another subgroup of nodes. This subgroup may continue to maintain a distributed ledger. In such scenario, the distributed ledgers of the subgroups of nodes evolve separately. A subgroup of devices that evolves separately from the original group is also denoted as a "fork" of the original group.

[0005] A connection between two separate distributed ledgers may form when nodes contributing to the distributed ledgers establish a connection. In such case, the distributed ledgers may be merged. However, it is not clear that there is a basis for a merger and sharing may reveal sensitive information.

[0006] For Internet-of-Things (IoT) applications, a large group of devices may operate a distributed ledger or blockchain. The consensus mechanism in such distributed ledger may be based on either a consensus algorithm or a mining process.

[0007] Although there exist distributed ledger techniques, it is generally desirable to make distributed ledger techniques more reliable and secure.

SUMMARY

[0008] According to a first aspect, the disclosure provides a method comprising determining if two separated distributed ledgers share a common history.

[0009] According to a further aspect, the disclosure provides a method comprising adapting the consensus mechanism of a distributed ledger change to the new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

[0010] According to a further aspect, the disclosure provides a system comprising one or more nodes that are configured to implement a distributed ledger and to determine if a separated distributed ledger shares a common history.

[0011] According to a further aspect, the disclosure provides a system comprising one or more nodes that are configured to implement a distributed ledger, the consensus mechanism of which depends on the current number of nodes available to the distributed ledger.

[0012] Further aspects are set forth in the dependent claims, the following description and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Embodiments are explained by way of example with respect to the accompanying drawings, in which:

Figs. 1a-e schematically describe a group of nodes that split into two subgroups and subsequently reestablish connection;

Fig. 2 schematically describes a method of an authentication process with which a first distributed ledger may decide to authorize merging with a second distributed ledger;

Figs. 3a-d schematically describe the splitting and rejoining of nodes and transactions in subgroups. In Fig. 3a, seven exemplary nodes A, B, C, D, E, F and G, during a timespan t1, are interconnected to each other so that they form a single group. In this timespan, nodes A-G record transactions to the same distributed ledger.

Figs. 4a-e schematically describe a second exemplifying group of nodes that split into two subgroups and subsequently reestablish connection;

Fig. 5 describes a process that may be implemented by a node group A when it loses and subsequently reestablishes connection to a node group B; and

Fig. 6 schematically describes an embodiment of an electronic device that may be used in context of the embodiments, e.g. as a node of a distributed ledger.

DETAILED DESCRIPTION OF EMBODIMENTS

[0014] In the embodiments described below, a local copy of a distributed ledger is stored on nodes accessing the distributed ledger. On a periodical basis, each node determines which group of nodes can be reached from that node. Connection between nodes may be established, terminated and reestablished. This may lead to subgroups of nodes that are in direct communication but which are not on direct communication with nodes of other subgroups. Subgroups of nodes may continue to record transactions on the distributed ledger. However, only transactions involving assets of the nodes included in a subgroup are considered valid. This effectively creates a fork of the distributed ledger for the subgroup.

[0015] In the embodiments, a method is disclosed comprising determining if two separated distributed ledgers share a common history. For example, if it is determined that two separated distributed ledgers share a common history, it may be concluded that the two distributed ledgers are forks of the same original distributed ledger.

[0016] A common history may for example relate to specific transactions or blocks of transactions that are stored in both distributed ledgers. In the case of blockchains, a common history may for example be reflected by historic blocks that two blockchains share.

[0017] Determining if two separated distributed ledgers share a common history may comprise using a challenge response authentication scheme. The challenge response authentication scheme may be configured to base challenges on the content of a distributed ledger. For example, as a challenge, a distributed ledger may be requested to return a hash of a block of the distributed ledger.

[0018] The distributed ledger to which the request is directed may then return the hash of the block of the distributed ledger. The distributed ledger that issued the request may check if the returned hash is correct, and establish that the two distributed ledgers share a common history based on one or more of such challenge requests.

[0019] The method may further comprise merging the two distributed ledgers if the determination has revealed that the two distributed ledgers are forks of the same original distributed ledger. For example, once a first group of nodes reestablishes a connection with a second group of nodes, the transactions that have occurred in the first group of nodes may be announced to the second group of nodes, and/or vice-versa.

[0020] Determining if two forks share a history may be carried out in the case that a communication between two forks of a distributed ledger is reestablished. The methods described in the embodiments may allow that distributed ledgers are merged based on their shared history without revealing privacy-sensitive information before the merge.

[0021] If the nodes of a distributed ledger are split up into several disconnected groups, as described above, this may lead to separated distributed ledgers of different sizes. Subgroups of devices may lose connection to the main distributed ledger for a substantial amount of time. However, these subgroups may want to continue maintaining a distributed ledger. The setting of a small group of devices may have implications for the reliability of the consensus approach used for the distributed ledger. For instance, when the distributed ledger uses a mining approach for consensus, a small group of devices may not have enough computational power to perform mining. On the other hand, running a consensus algorithm may not be adequate in case a large majority of the subgroup is controlled by a single entity.

[0022] Accordingly, the embodiments also disclose a method comprising adapting the consensus mechanism of a distributed ledger change to the new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes. A consensus mechanism may for example be based on a proof of work (e.g. a mining process), or it may be based on a consensus algorithm such as a Byzantine fault tolerance algorithm, or the like.

[0023] For example, the consensus mechanism of a distributed ledger may be adapted in the case that the number of nodes contributing to a distributed ledger changes from a large group of nodes to a small group of nodes.

[0024] Adapting the consensus mechanism may comprise switching from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm. For example, when a large group of nodes, which uses mining to achieve consensus, is split up into smaller groups of nodes, and a smaller group of nodes does not have enough computational power to perform mining, the consensus mechanism may be switched to a consensus algorithm such as the Byzantine fault tolerance algorithm.

[0025] Alternatively, adapting the consensus mechanism may comprise lowering the complexity of a mining process. For example, when a large group of nodes, which uses mining to achieve consensus, is split up into smaller groups of nodes, and a smaller group of nodes does not have enough computational power to perform mining at the complexity defined in the original distributed ledger, the complexity of the mining process may be lowered.

[0026] Once a small group of nodes reestablishes a connection with a large group of nodes, the transactions that have occurred may be announced to the large group of nodes as it was described above. The overall set of nodes may then incorporate these transactions and other changes that have occurred.

[0027] The embodiments thus may provide a mechanism for a group of nodes that loses connection from a distributed ledger to continue using its distributed ledger in a feasible manner. Once connection is reestablished, any transaction can be announced and incorporated into the overall distributed ledger.

[0028] The embodiments also disclose a system comprising one or more nodes that are configured to implement a distributed ledger and to determine if two separated distributed ledgers share a common history.

[0029] The embodiments further disclose a system comprising multiple nodes that are configured to implement a distributed ledger the consensus mechanism of which depends on the current number of nodes available to the distributed ledger.

[0030] A node of a distributed ledger may be any electronic device, e.g. a personal computer, a work station, a mobile computing device such as a smartphone, a tablet computer, or the like. An electronic device that acts as node of a distributed ledger may for example comprise a CPU, a storage unit (e.g. a hard drive or SSD), a memory unit (e.g. a RAM), input/output interfaces such as an Ethernet interface, a WiFi interface or the like, and user interfaces such as a keyboard, a display, a loudspeaker, and/or a microphone.

[0031] Figs. 1a-e schematically describe a first exemplifying group of nodes that split into two subgroups and subsequently reestablish connection.

[0032] In Fig. 1a, a group A+B comprises multiple nodes that each contribute to a shared distributed ledger. Some of the nodes are in direct connection with each other. The nodes may for example be interconnected by a LAN or WAN network, or by other communication technologies. All nodes of group A+B are at least in indirect connection with each other so that they all can share the same distributed ledger. The nodes record transactions on the shared distributed ledger using a predefined consensus mechanism, as it is known to the skilled person as blockchain technology. In this embodiment, a local copy of the shared distributed ledger is stored on each node accessing the distributed ledger. On a periodical basis, each node determines which group of nodes can be reached from that node.

[0033] Connection between nodes may be established, terminated and reestablished. This may lead to subgroups of nodes that are in direct communication but which are not in direct communication with nodes of other subgroups.

[0034] In Fig. 1b, it is shown that two devices N_A and N_B of group A+B lose their direct connection. This results in that the nodes of group A+B are separated into two subgroups A and B which are no longer interconnected to each other, as it is shown in Fig. 1c.

[0035] According to Fig. 1c, the nodes of group A and the nodes of group B can no longer contribute to the same distributed ledger. The subgroups of nodes may, however, continue to record transactions on the distributed ledger. However, only transactions involving assets of the nodes included in a subgroup are considered valid. This effectively creates a fork of the distributed ledger for the subgroup. I.e. each group of nodes may continue to record transactions into the respective distributed ledger they maintain, which results in two separated distributed ledgers (forks) that share the same history but that evolve in different ways.

[0036] In Fig. 1d, it is shown that two devices N_A and N_B of groups A and B establish a direct connection so that the two subgroups A and B reestablish connection. This results in that the nodes can again contribute to a single distributed ledger. To this end, the distributed ledger of group A and the distributed ledger of group B may merge as is described below in more detail. Fig. 1e finally describes the situation in which the groups A and B are rejoined as group A+B. The nodes again contribute to a common shared distributed ledger.

[0037] As shown above, two distributed ledgers that share the same history may evolve independently over time after a fork has occured. Once nodes of two separated distributed ledgers establish a connection, it may make sense to merge their respective content. However, simply sharing the distributed ledgers may reveal sensitive and private information.

[0038] According to the embodiments described below in more detail, the content of a distributed ledger may be used to construct an authentication scheme with which it can be decided if two distributed ledgers should be merged. A process that may be implemented by a distributed ledger to implement such authentication process is now further described with reference to Fig. 2.

[0039] Fig. 2 schematically describes a method of an authentication process with which a first distributed ledger may decide to authorize merging with a second distributed ledger. At 201, a first distributed ledger called distributed ledger A creates a set of L challenges $CA_1,...,CA_L$. For instance each of the challenges CA_i may request a hash of block i to be returned. At 202, the distributed ledger A sends the challenges $CA_1,...,CA_L$ to the second distributed ledger B. At 203, distributed ledger B responds to each of the challenges. Distributed ledger B computes the responses $H(RA_1),...,H(RA_L)$ for each of the challenges and distributed ledger B returns the responses $H(RA_1),...,H(RA_L)$, where $H(RA_i)$ denotes a hash function of the response RA_i. According to this embodiment, the challenge is based on the transactions present in the distributed ledger. The general idea behind this embodiment is that distributed ledgers share a common history if they have copies of the same transactions (or block of transactions). At 204, distributed ledger A verifies the responses H(RA_1),...,H(RA_L). At 205, distributed ledger A decides to authorize a merge with distributed ledger B if the responses H(RA_1),...,H(RA_L) are positively validated, e.g. if the number of correct responses exceeds a predefined number *K*. If the responses H(RA_1),...,H(RA_L) are not positively validated, the process continues at 207, i.e. the process ends. If the responses H(RA_1),...,H(RA_L) are positively validated, the process continues at 206. At 206, distributed ledger A authorizes sharing its content (e.g. transactions or blocks of transactions) with distributed ledger B. The authorization process then ends at 207. After authorization, distributed ledger A may share its content with distributed ledger B, as it is disclosed below in more detail.

[0040] Figs. 3a-d schematically describe the splitting and rejoining of nodes and transactions in subgroups. In Fig. 3a, seven exemplary nodes A, B, C, D, E, F and G, during a timespan t1, are interconnected to each other so that they form a single group. In this timespan, nodes A-G record transactions to the same distributed ledger.

[0041] Fig. 3a depicts two exemplary transactions T1 and T2 that are recorded to the same distributed ledger. As also depicted in Fig. 3a, each node A-G holds an own local copy of the shared distributed ledger. That is, each of the nodes A-G holds a copy of exemplary transactions T1 and T2.

[0042] As shown in Fig. 3b, after the elapse of timespan t1, the nodes A-G split into two separated subgroups A, B, C and D, E, F, G. During timespan t2 that follows after timespan t1, the two subgroups continue to record transactions to their respective distributed ledger. However, as the two subgroups are disconnected from each other, these transactions are not shared between the respective distributed ledgers. That is, the distributed ledgers of subgroup A, B, C and subgroup D, E, F, G, even though sharing the same history (transactions T1 and T2), evolve differently. Here, for example, subgroup A, B, C records a transaction T3, whereas subgroup D, E, F, G records a transaction T4.

[0043] As shown in Fig. 3c, after the elapse of timespan t2, the nodes reconfigure to three new subgroups. A first subgroup comprises nodes A, B, D, a second subgroup comprises nodes C, E, F and a third subgroup comprises node G. Before the elapse of timespan t2, nodes A, B and node D belonged to different distributed ledgers. When reestablishing contact, they validate that they share a common history (transactions T1 and T2) and decide to merge their distributed ledgers. This results in that the nodes A, B and D exchange their knowledge about transactions T3 and T4 so that the resulting merged distributed ledger comprises both transactions, T3 and T4, in addition to the transactions T1 and T2 that form a common history of both distributed ledgers. The same applies to nodes C, E, and F. Before the elapse of timespan t2, nodes C, E and node F belonged to different distributed ledgers. When reestablishing contact, they validate that they share a common history (transactions T1 and T2) and decide to merge their distributed ledgers. This results in that the nodes C, E, and F exchange their knowledge about transactions T3 and T4 so that the resulting merged distributed ledger comprises both transactions, T3 and T4, in addition to the transactions T1 and T2 that form a common history of both distributed ledgers. Node G, to the contrary, after timespan t2, splits off to form its own subgroup and, accordingly, does not make contact with any other node. During timespan t3, subgroup A, B, D records a transaction T5, subgroup C, E, F records a transaction T6, and node G records a transaction T7.

[0044] As shown in Fig. 3d, after the elapse of timespan t3, the nodes reestablish connection and form to the original group A-G. When reestablishing contact, the nodes A, B, C, D, E, and F validate that they share a common history (transactions T1, T2, T3, T4) and decide to merge their distributed ledgers. This results in that nodes A, B, C, D, E, and F exchange their knowledge about transactions T5 and T6 so that the resulting merged distributed ledger comprises both transactions, T5 and T6, in addition to the transactions T1, T2, T3 and T4 that form a common history of the previous distributed ledgers. However, as node G contains only transactions T1, T2, T4, and misses transaction T3 in its history, validating the history of node G will fail. Node G is thus not authorized to share its content with the remaining nodes. Node G is, however, free to dismiss its own history and continue contributing to the merged distributed ledger established by nodes A, B, C, D, E, and F, which will effectively result in a loss of transaction T7.

[0045] Figs. 4a-e schematically describe a second exemplifying group of nodes that split into two subgroups and subsequently reestablish connection. The example of Figs. 4a-e substantially corresponds to the example of Figs. 2a-e, however, group A that splits of group A+B is a very small group that consists only of three nodes. Initially, all nodes are in communication through e.g. a network, and maintain a distributed ledger or blockchain. At some point in time node group A gets separated from node group B. Each of the subgroups continues to maintain the distributed ledger. However, the number of devices in the subgroups of devices has changed substantially. In case the original distributed ledger uses a mining process, the number of devices in subgroup B may not be enough to provide enough computational power to perform mining.

[0046] In the embodiment described below in more detail, this may be solved in two ways. First, the complexity of the mining operation may be lowered by a factor that corresponds to the factor of reduction in computational power. In such a way, the mining process for a subgroup of devices is able to finish in a timely manner. Second, the mining process may be switched to a consensus algorithm such as the Byzantine fault tolerance protocol.

[0047] Disconnected subgroups continue to maintain a local distributed ledger or blockchain. Once a subgroup reestablishes connection to a larger group of devices, the transaction incorporated in the distributed ledger of the subgroup may be announced to the larger group and incorporated. The latter may be performed by the original consensus mechanism of the large group of devices.

[0048] A process that may be implemented once device group A loses its connection to device group B is now further described with reference to Fig. 5.

[0049] Fig. 5 describes a process that may be implemented by device group A when it loses and subsequently reestablishes connection to device group B.

[0050] At 501, the nodes of group A (distributed ledger A) determine the cumulative mining capabilities. At 502, the nodes of distributed ledger A determine the expected mining time based on the cumulative mining capabilities determined at 501. At 503, the nodes of distributed ledger A determine whether the cumulative mining capabilities are sufficient to support adding new transactions to the distributed ledger within acceptable time. If the cumulative mining capabilities are sufficient to support adding new transactions to the distributed ledger within acceptable time, the process continues at 505. Otherwise, the process continuous at 504. At 504, the nodes in group A switch to a consensus mechanism. At 505, the nodes in group A keep their original consensus mechanism. At 506, nodes in group A continue adding transactions to the distributed ledger. During this time, the nodes act as an independent group and maintain a distributed ledger.

[0051] When a connection to node group B is reestablished, device group A may announce the transactions that have occurred to device group B and these transactions may be incorporated into the overall distributed ledger that is shared between node group A and node group B. In a similar way, node group B may announce transactions that are also incorporated in the distributed ledger to node group A.

[0052] The following table provides an example configuration of adapting a consensus algorithm:

Number **N of nodes in subgroup**	**Consensus mechanism**
100 < N	Mining (proof of work)
50 < N ≤ 100	Byzantine fault tolerance with 51% consensus required
20 < N ≤ 50	Byzantine fault tolerance with 81% consensus required
0 < N ≤ 20	Byzantine fault tolerance with 91% consensus required

[0053] It has been described above that nodes of a distributed ledger may be represented by electronic devices.

[0054] Fig. 6 schematically describes an embodiment of an electronic device 600 that may be used in context of the embodiments, e.g. as a node of a distributed ledger. The electronic device 600 comprises a CPU 601 as processor. The electronic device 600 further comprises a microphone 610, a loudspeaker 611, a display 612, and a keyboard 613 that are connected to the processor 601. These units 610, 611, 612, and 613 act as a man-machine interface and enable a dialogue between a user and the electronic device. The electronic device 600 further comprises an Ethernet interface 604 and a WiFi interface 605. These units 604, 605 act as I/O interfaces for data communication with external devices such as other nodes of a distributed ledger. The electronic device 600 further comprises a data storage 602 (e.g. a Hard Drive, Solid State Drive, or SD card) and a data memory 603 (e.g. a RAM). The data memory 603 is arranged to temporarily store or cache data or computer instructions for processing by processor 601. The data storage 602 is arranged as a long-term storage, e.g. for recording transactions in a blockchain.

[0055] It should be noted that the description above is only an example configuration. Alternative configurations may be implemented with additional or other sensors, storage devices, interfaces or the like. For example, in alternative embodiments, WiFi interface 605, microphone 610, display 612, and/or loudspeaker 611, or keyboard 613 may be omitted or replaced by other units.

[0056] The skilled person will readily appreciate that in so far it is described in the embodiments that a distributed ledger is performing some activity; it is generally understood that the nodes, i.e. the electronic devices that constitute the network, perform this action either in cooperation, or as subgroups of all devices, or as single devices. For example, a distributed ledger may create a set of challenges (e.g. 201 in Fig. 2) by configuring a single node (electronic device) to create the challenges. In other embodiments, multiple or all of the nodes contributing to the distributed ledger may be configured to perform the action. To this end, the nodes communicate with each other as known in the art of distributed ledger technology. The creation of challenges, the sending of challenges and the verifying of challenge responses may but need not necessarily be carried out by one or more full nodes of the network.

[0057] It should be recognized that the embodiments describe methods with an exemplary order of method steps. The specific order of method steps is, however, given for illustrative purposes only and should not be construed as binding. For example, the order of 501 and 503 in the embodiment of Fig. 5 may be exchanged. Other changes of the order of method steps may be apparent to the skilled person.

[0058] It should further be recognized that the division of the electronic device 600 into units 601 to 613 is only made for illustration purposes and that the present disclosure is not limited to any specific division of functions in specific units. For instance, the electronic device 600 could be implemented by a respective programmed processor, field programmable gate array (FPGA) and the like.

[0059] The methods disclosed here can also be implemented as a computer program causing a computer and/or a processor (such as CPU 601 in Fig. 6), to perform the methods when being carried out on the computer and/or processor. In some embodiments, also a non-transitory computer-readable recording medium is provided that stores therein a computer program product, which, when executed by a processor, such as the processor described above, causes the method described to be performed.

[0060] All units and entities described in this specification and claimed in the appended claims can, if not stated otherwise, be implemented as integrated circuit logic, for example on a chip, and functionality provided by such units and entities can, if not stated otherwise, be implemented by software.

[0061] In so far as the embodiments of the disclosure described above are implemented, at least in part, using a software-controlled data processing apparatus, it will be appreciated that a computer program providing such software control and a transmission, storage or other medium by which such a computer program is provided are envisaged as aspects of the present disclosure.

[0062] Note that the present technology can also be configured as described below.

(1) A method comprising determining if two separated distributed ledgers share a common history.

(2) The method of (1), wherein determining if two separated distributed ledgers share a common history comprises using a challenge response authentication scheme.

(3) The method of (2), wherein the challenge response authentication scheme is configured to base challenges on the content of a distributed ledger.

(4) The method of (2) or (3), wherein, as a challenge, a distributed ledger is requested to return a hash of a block of the distributed ledger.

(5) The method of anyone of (1) to (4), further comprising merging the two distributed ledgers if the determination has revealed that the two distributed ledgers are forks of the same original distributed ledger.

(6) The method of anyone of (1) to (5), in which the determining if two forks share a common history is carried out in the case that a communication between two forks of a distributed ledger is reestablished.

(7) A method comprising adapting the consensus mechanism of a distributed ledger to a new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

(8) The method of (7), wherein adapting the consensus mechanism comprises switching from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm.

(9) The method of (7) or (8), wherein adapting the consensus mechanism comprises lowering the complexity of a mining process.

(10) The method of anyone of (1) to (6) comprising adapting the consensus mechanism of a distributed ledger to a new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

(11) The method of anyone of (1) to (6), wherein adapting the consensus mechanism comprises switching from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm.

(12) The method of anyone of (1) to (6), wherein adapting the consensus mechanism comprises lowering the complexity of a mining process.

(13) A system comprising one or more nodes that are configured to implement a distributed ledger and to determine if a separated distributed ledger shares a common history.

(14) The system of (13), wherein the nodes are configured to use a challenge response authentication scheme to determine if the separated distributed ledger shares a common history.

(15) The system of (14), wherein the challenge response authentication scheme is configured to base challenges on the content of a distributed ledger.

(16) The system of (14) or (15), wherein, as a challenge, the separated distributed ledger is requested to return a hash of a block of the separated distributed ledger.

(17) The system of anyone of (13) to (16), wherein the one or more nodes are configured to merge the distributed ledger and the separated distributed ledger if the determination has revealed that the two distributed ledgers are forks of the same original distributed ledger.

(18) The system of anyone of (13) to (17), wherein the one or more nodes are configured to determine if two forks share a history in the case that a communication between two forks of a distributed ledger is reestablished.

(19) A system comprising one or more nodes that are configured to implement a distributed ledger, the consensus mechanism of which depends on the current number of nodes available to the distributed ledger.

(20) The system of (19), wherein the one or more nodes are configured to adapt the consensus mechanism of the distributed ledger to the new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

(21) The system of (19) or (20), wherein the one or more nodes are configured to switch from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm in order to adapt the consensus mechanism.

(22) The system of (19) to (21), wherein the one or more nodes are configured to lower the complexity of a mining process in order to adapt the consensus mechanism.

(23) Electronic device comprising a processor configured to determine if two separated distributed ledgers share a common history.

(24) Electronic device comprising a processor configured to adapt the consensus mechanism of a distributed ledger to a new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

(25) Electronic device comprising a processor configured to implement the method of anyone of (1) to (12).

(26) A computer program comprising program code causing a computer to perform the method according to anyone of (1) to (12), when being carried out on a computer.

(27) A non-transitory computer-readable recording medium that stores therein a computer program product, which, when executed by a processor, causes the method according to anyone of (1) to (12) to be performed.

Claims

1. A method comprising determining if two separated distributed ledgers share a common history.

2. The method of claim 1, wherein determining if two separated distributed ledgers share a common history comprises using a challenge response authentication scheme.

3. The method of claim 2, wherein the challenge response authentication scheme is configured to base challenges on the content of a distributed ledger.

4. The method of claim 2, wherein, as a challenge, a distributed ledger is requested to return a hash of a block of the distributed ledger.

5. The method of claim 1, further comprising merging the two distributed ledgers if the determination has revealed that the two distributed ledgers are forks of the same original distributed ledger.

6. The method of claim 1, in which the determining if two forks share a common history is carried out in the case that a communication between two forks of a distributed ledger is reestablished.

7. A method comprising adapting the consensus mechanism of a distributed ledger to a new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

8. The method of claim 7, wherein adapting the consensus mechanism comprises switching from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm.

9. The method of claim 7, wherein adapting the consensus mechanism comprises lowering the complexity of a mining process.

10. A system comprising one or more nodes that are configured to implement a distributed ledger and to determine if a separated distributed ledger shares a common history.

11. The system of claim 10, wherein the nodes are configured to use a challenge response authentication scheme to determine if the separated distributed ledger shares a common history.

12. The system of claim 11, wherein the challenge response authentication scheme is configured to base challenges on the content of a distributed ledger.

13. The system of claim 11, wherein, as a challenge, the separated distributed ledger is requested to return a hash of a block of the separated distributed ledger.

14. The system of claim 10, wherein the one or more nodes are configured to merge the distributed ledger and the separated distributed ledger if the determination has revealed that the two distributed ledgers are forks of the same original distributed ledger.

15. The system of claim 10, wherein the one or more nodes are configured to determine if two forks share a history in the case that a communication between two forks of a distributed ledger is reestablished.

16. A system comprising one or more nodes that are configured to implement a distributed ledger the consensus mechanism of which depends on the current number of nodes available to the distributed ledger.

17. The system of claim 16, wherein the one or more nodes are configured to adapt the consensus mechanism of the distributed ledger to the new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

18. The system of claim 17, wherein the one or more nodes are configured to switch from mining to a consensus algorithm such as the Byzantine fault tolerance algorithm in order to adapt the consensus mechanism, or wherein the one or more nodes are configured to lower the complexity of a mining process in order to adapt the consensus mechanism.

19. Electronic device comprising a processor configured to determine if two separated distributed ledgers share a common history.

20. Electronic device comprising a processor configured to adapt the consensus mechanism of a distributed ledger to a new size of the distributed ledger in the case that the number of nodes contributing to the distributed ledger changes.

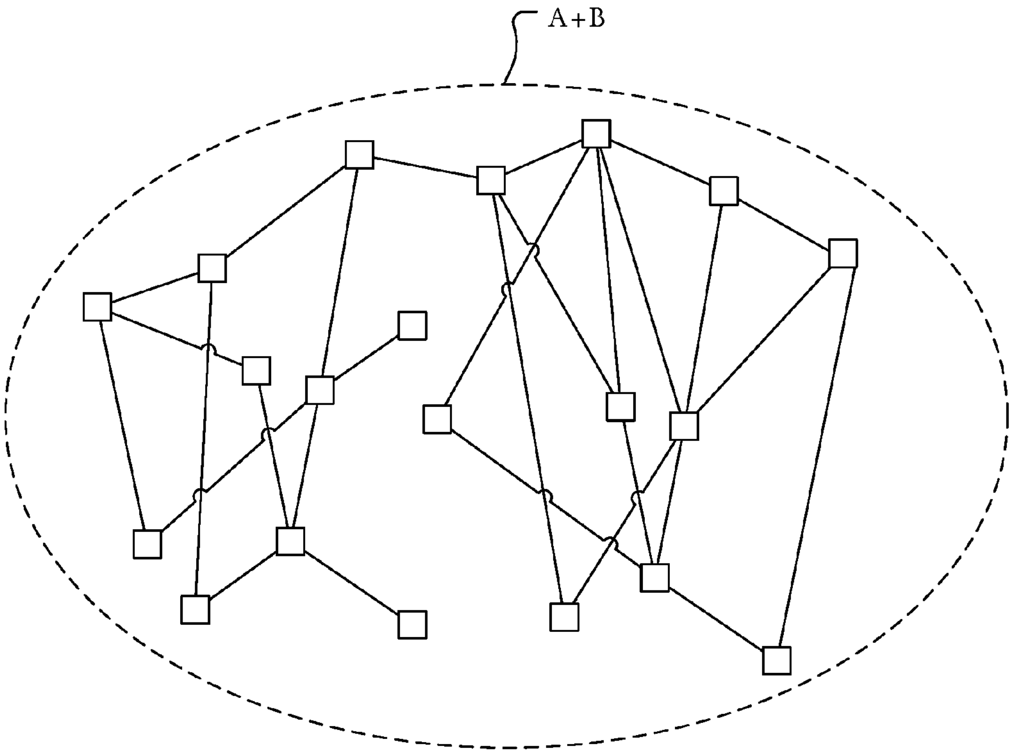

Fig. 1a

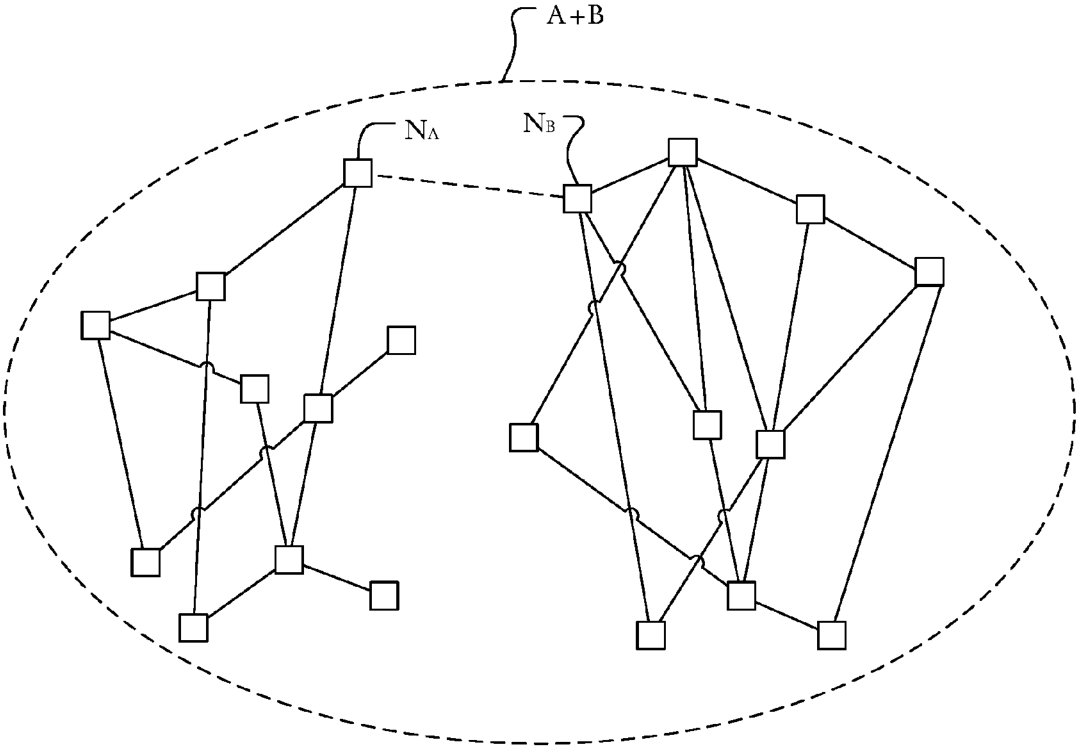

Fig. 1b

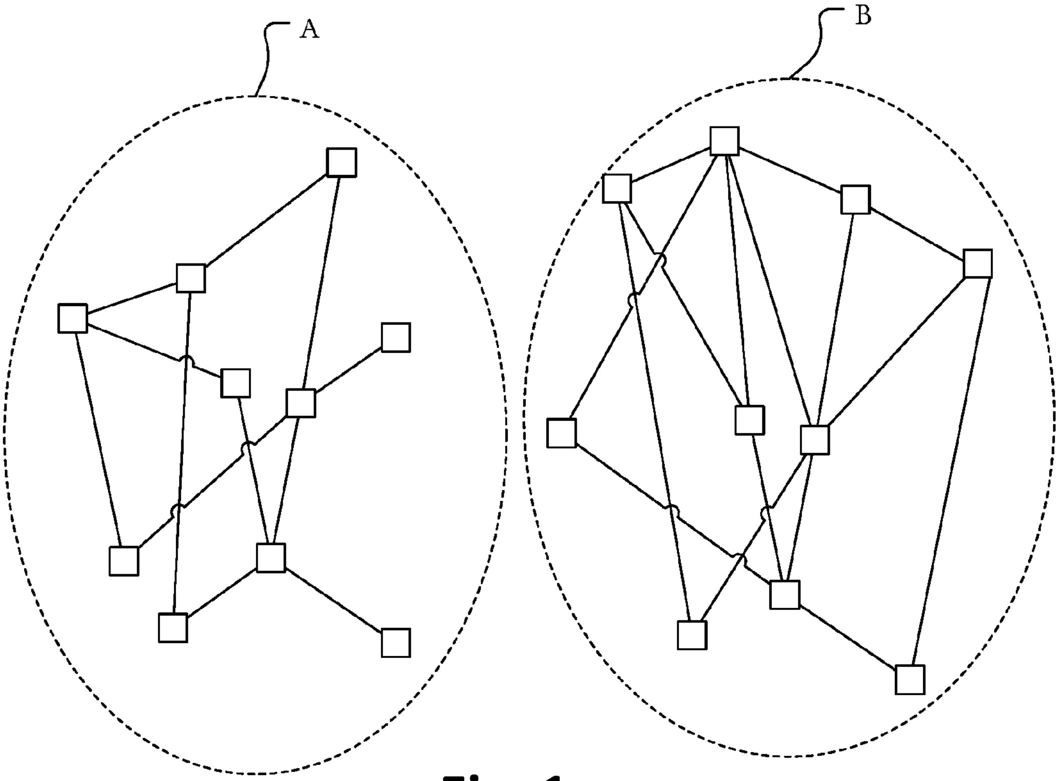

Fig. 1c

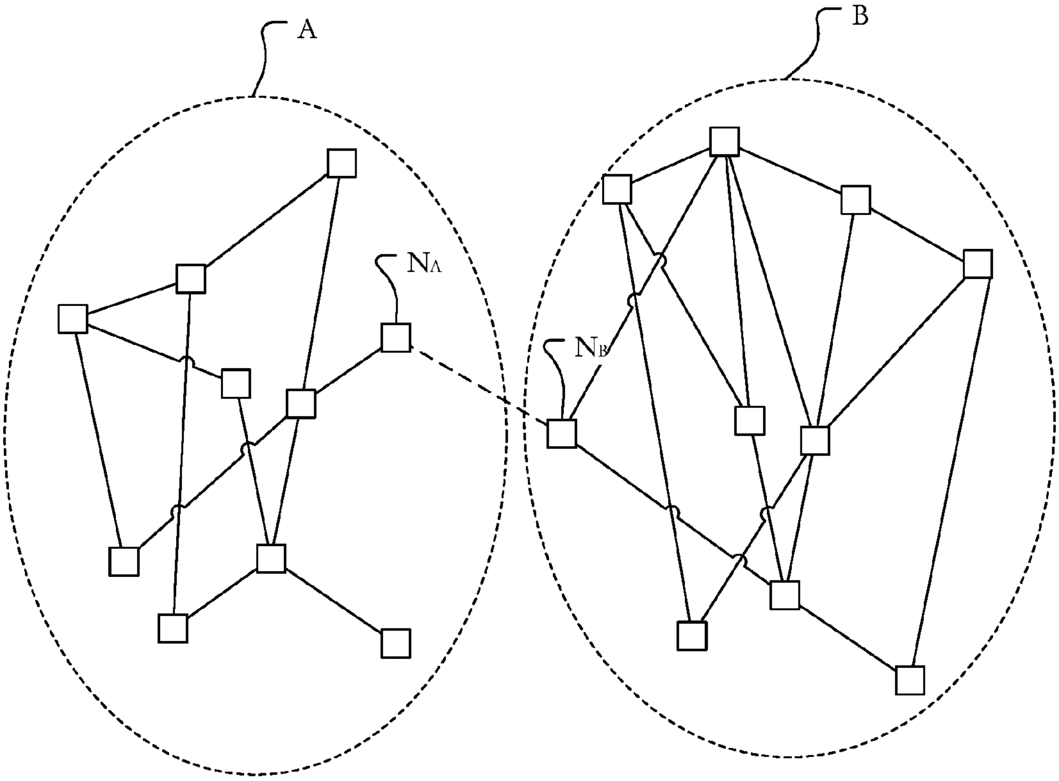

Fig. 1d

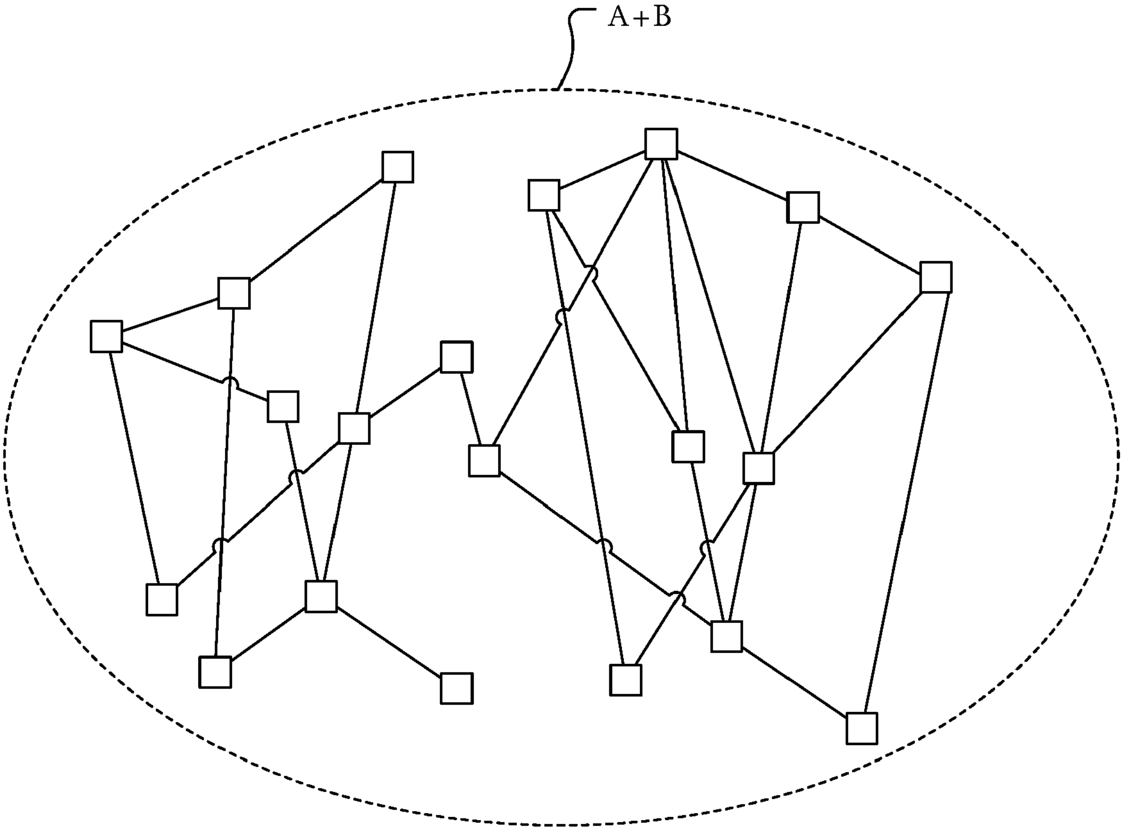

Fig. 1e

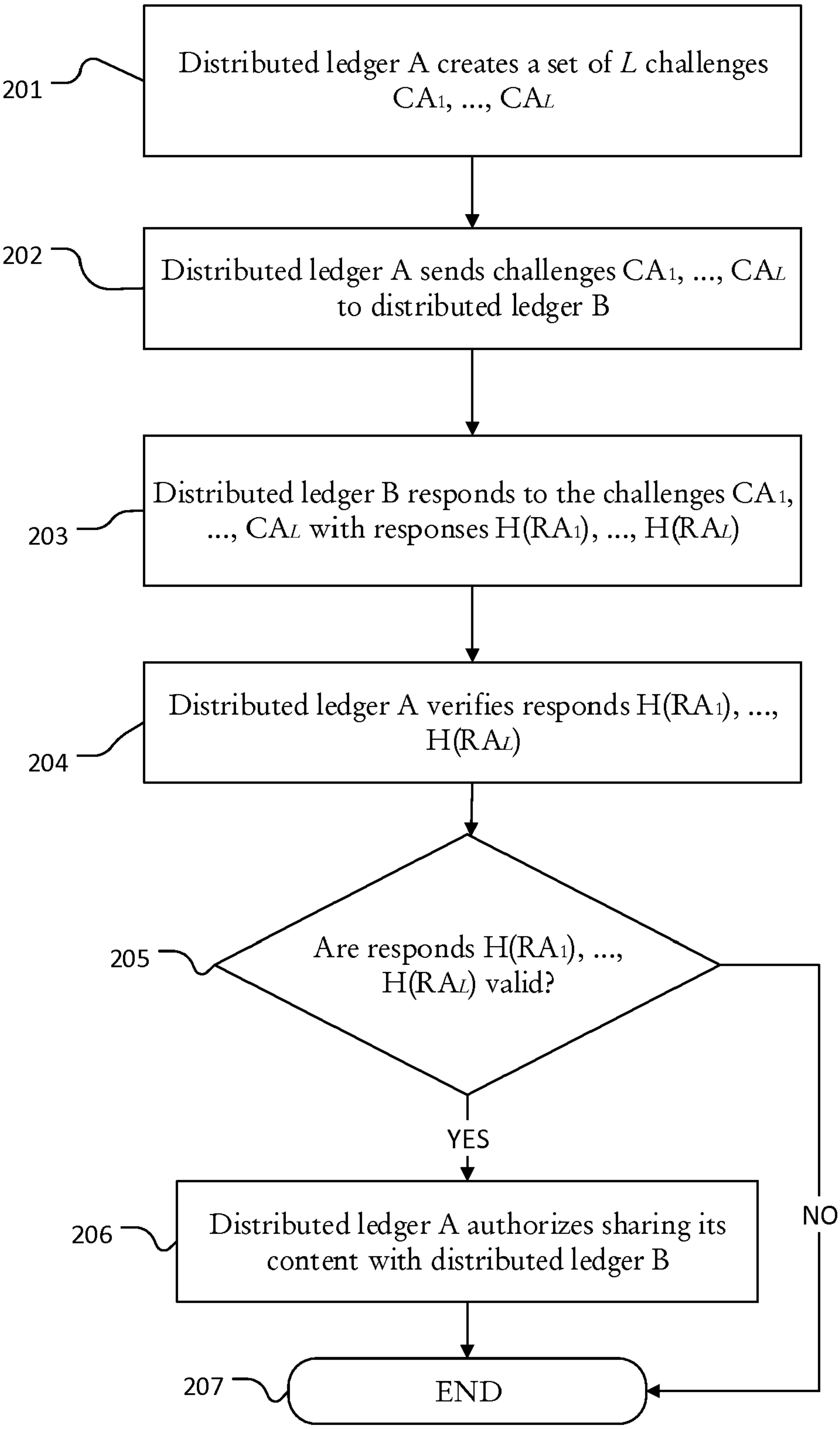
201
Distributed ledger A creates a set of L challenges CA1, ..., CAL
202
Distributed ledger A sends challenges CA1, ..., CAL to distributed ledger B
203
Distributed ledger B responds to the challenges CA1, ..., CAL with responses H(RA1), ..., H(RAL)
204
Distributed ledger A verifies responds H(RA1), ..., H(RAL)
205
Are responds H(RA1), ..., H(RAL) valid?
YES
NO
206
Distributed ledger A authorizes sharing its content with distributed ledger B
207
END

Fig. 2

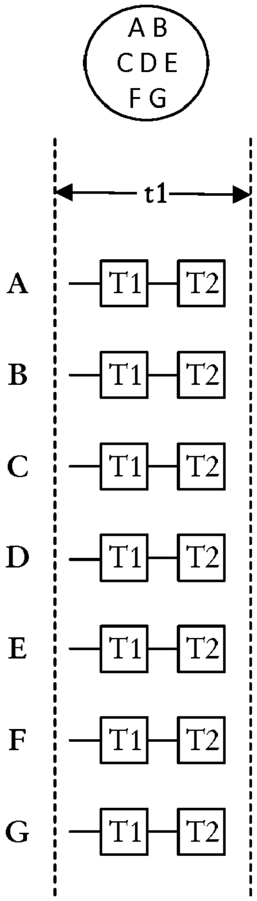

Fig. 3a

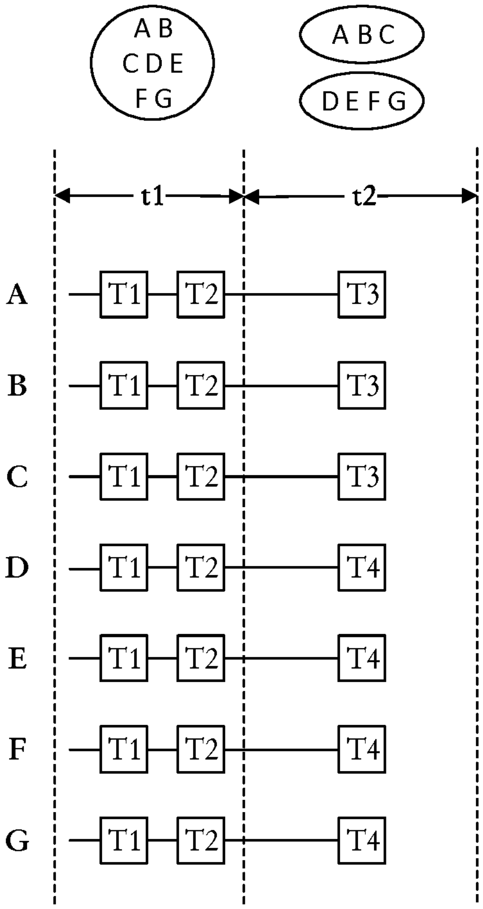

Fig. 3b

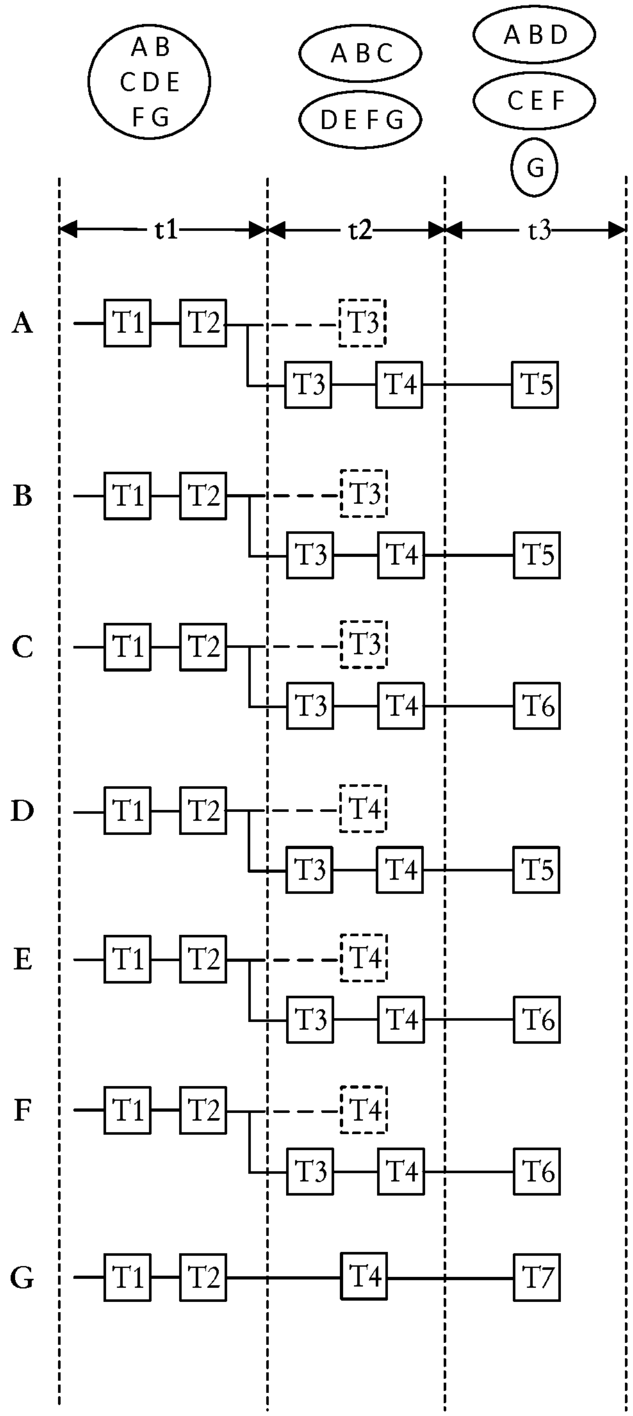
A B
C D E
F G
A B C
D E F G
A B D
C E F
G
t1
t2
t3
A
T1
T2
T3
T3
T4
T5
B
T1
T2
T3
T3
T4
T5
C
T1
T2
T3
T3
T4
T6
D
T1
T2
T4
T3
T4
T5
E
T1
T2
T4
T3
T4
T6
F
T1
T2
T4
T3
T4
T6
G
T1
T2
T4
T7

Fig. 3c

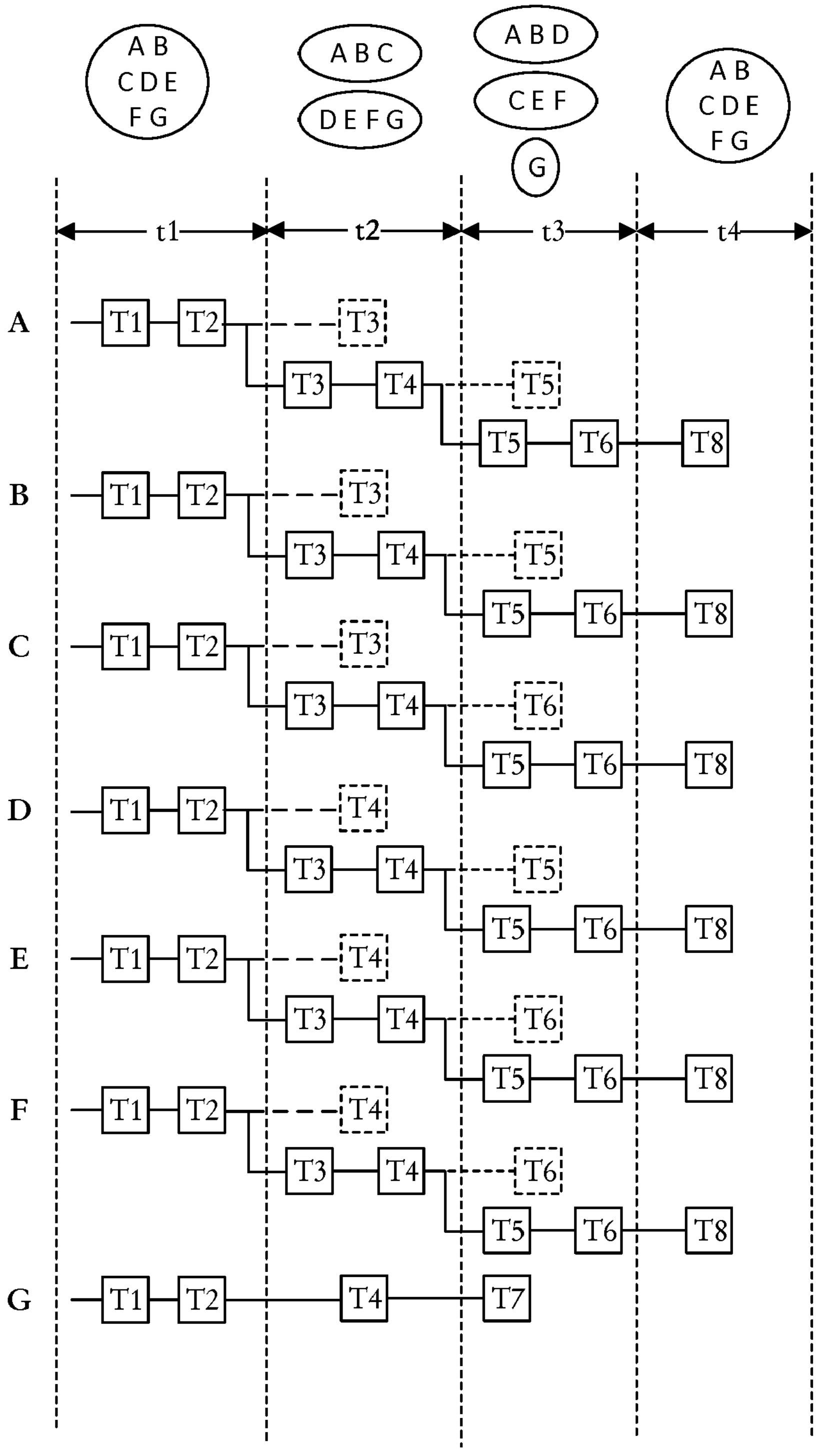
A B
C D E
F G
A B C
D E F G
A B D
C E F
G
A B
C D E
F G
t1
t2
t3
t4
A
B
C
D
E
F
G
T1
T2
T3
T4
T5
T6
T7
T8

Fig. 3d

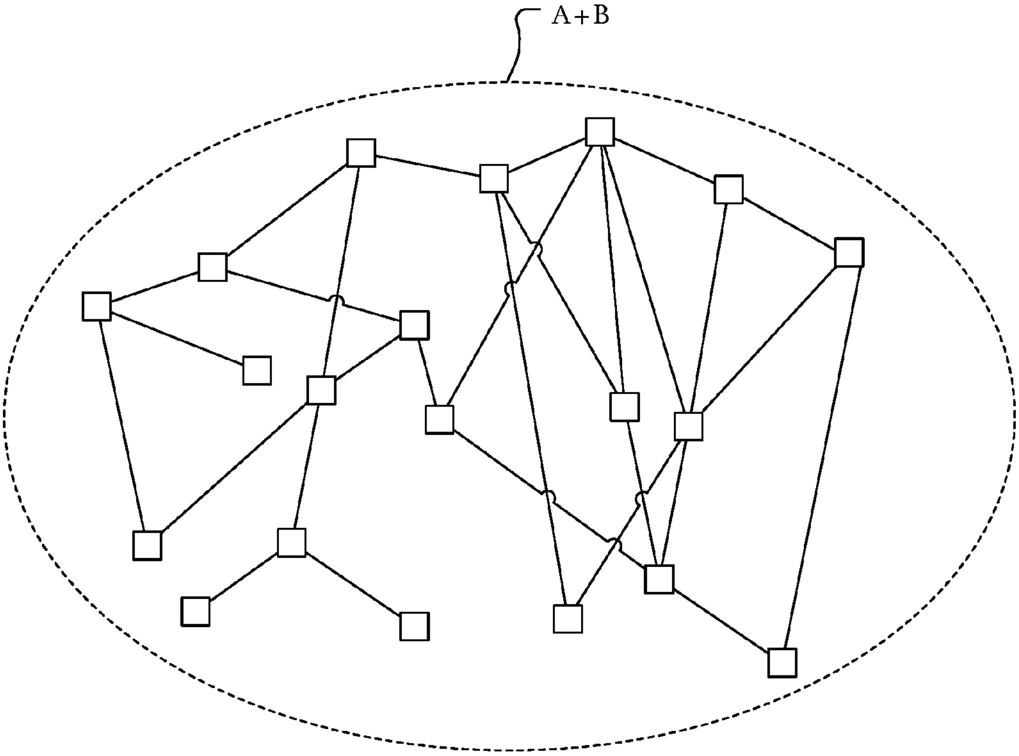

Fig. 4a

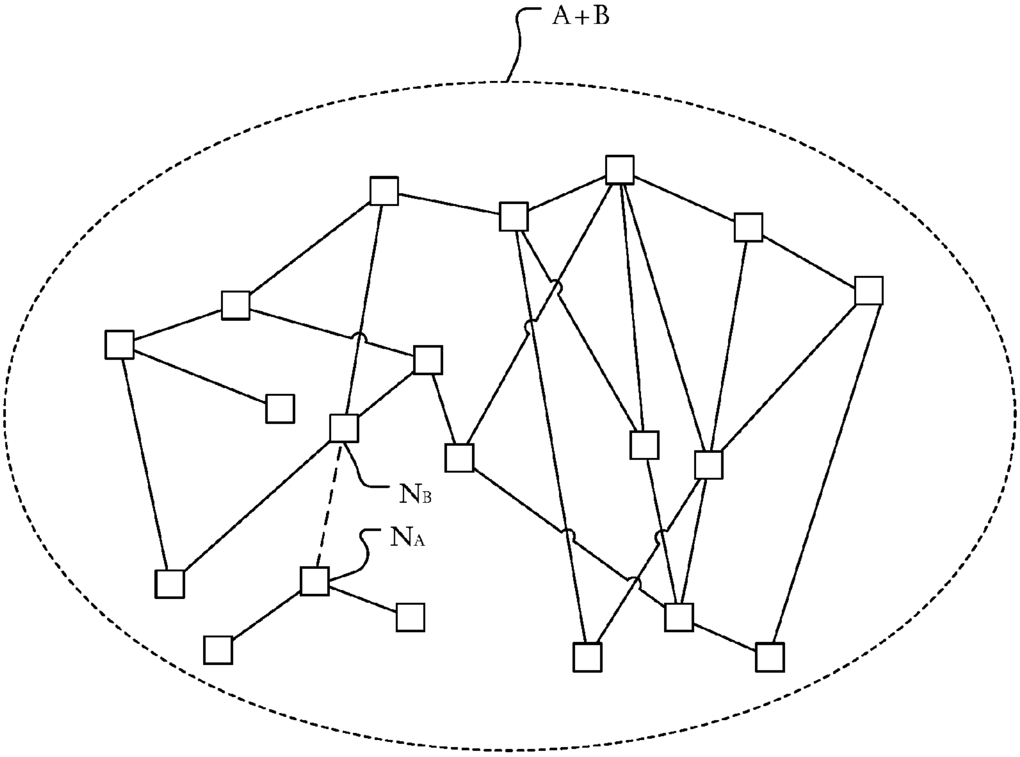

Fig. 4b

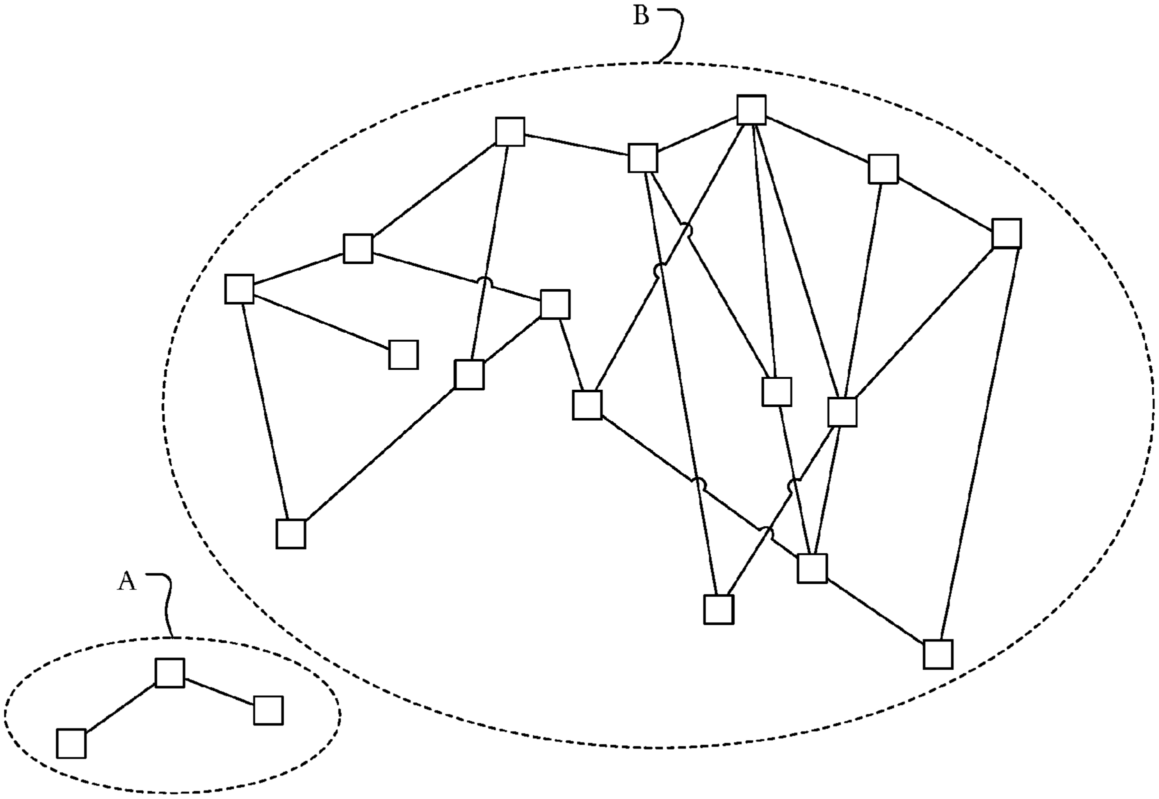

Fig. 4c

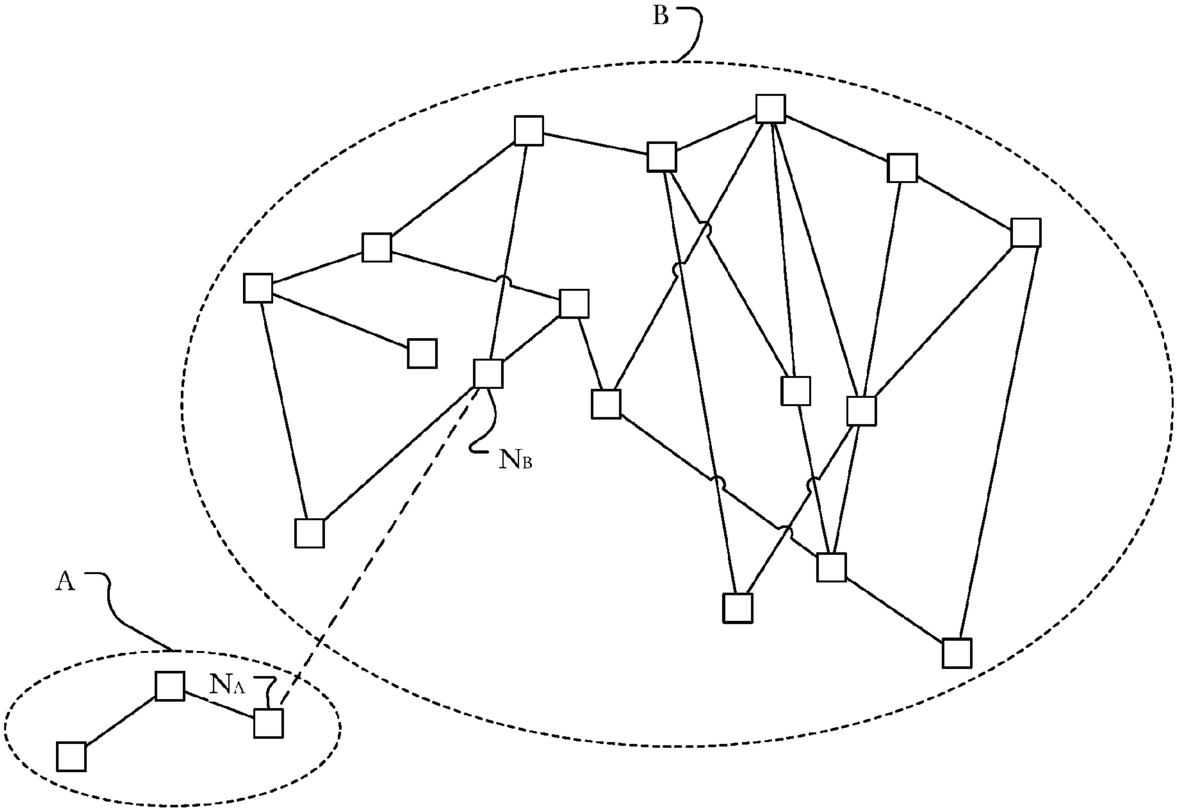

Fig. 4d

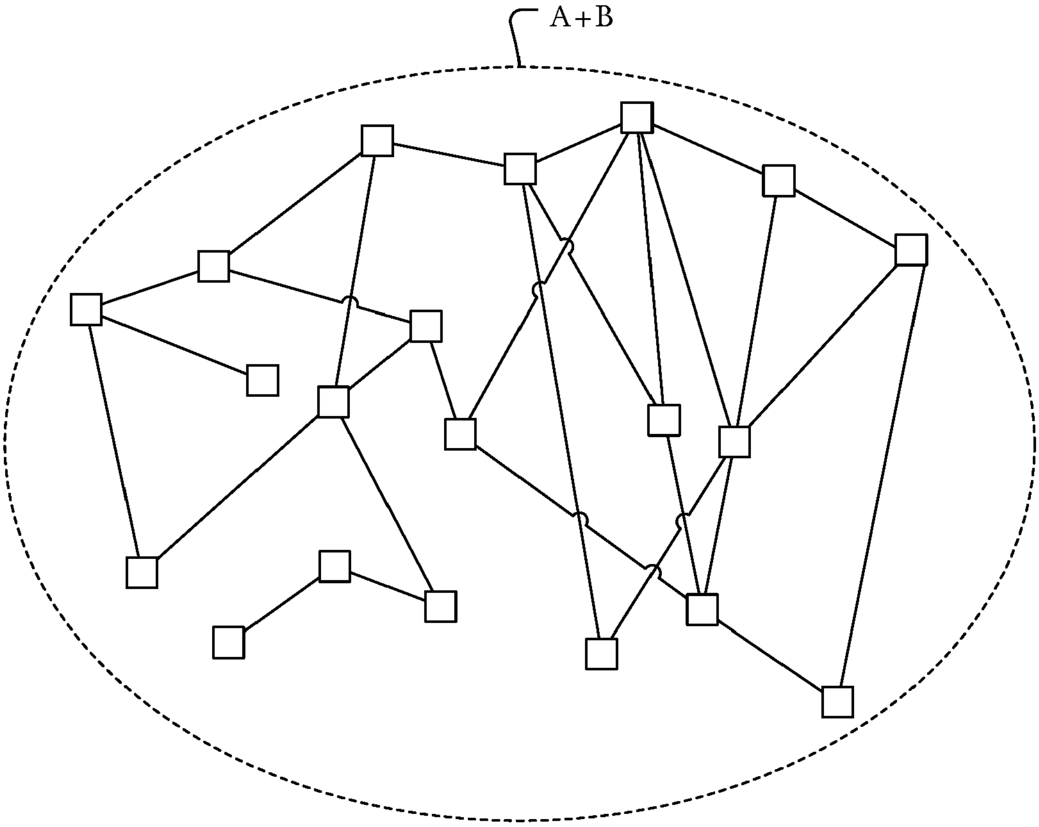

Fig. 4e

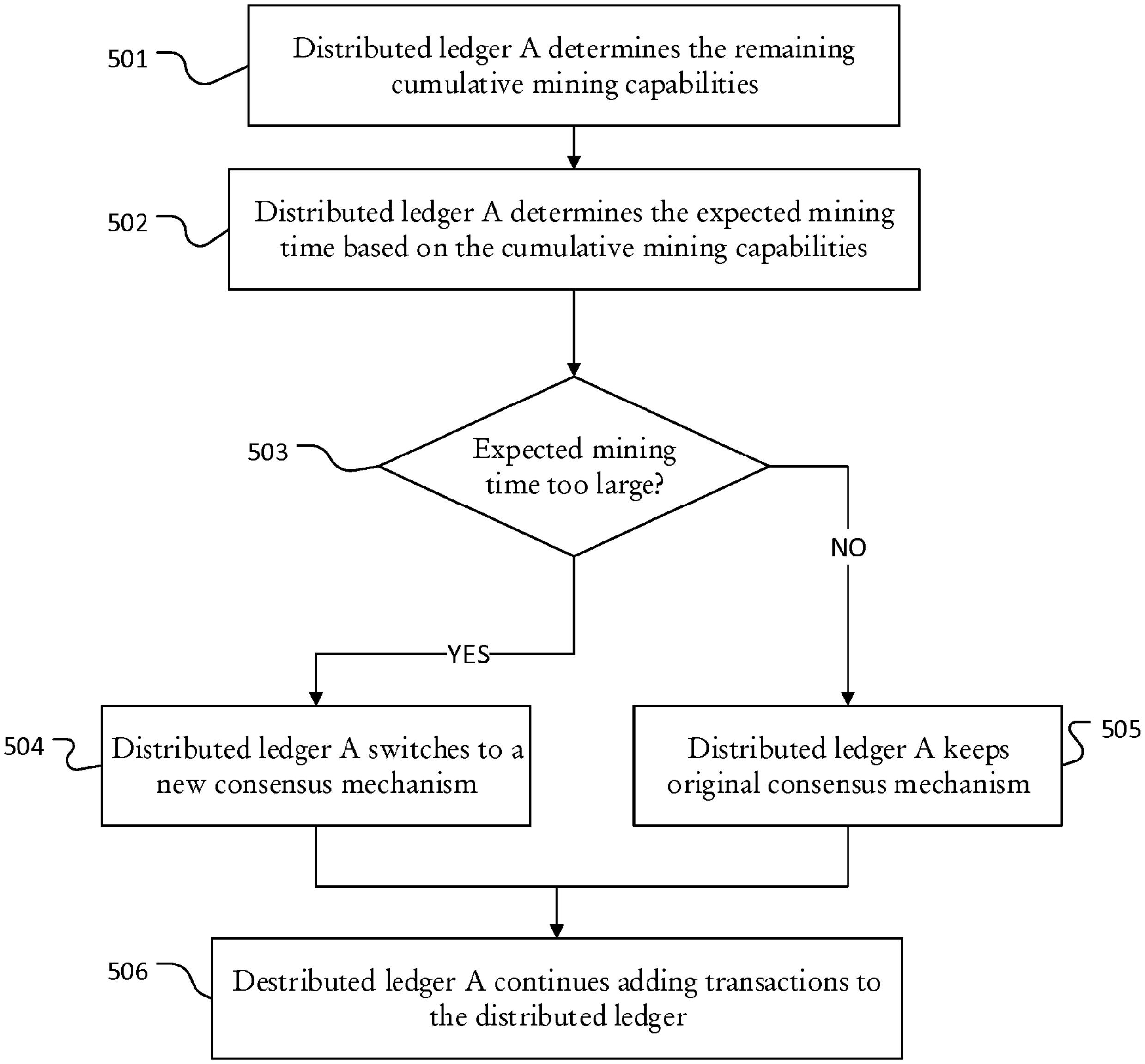
501
Distributed ledger A determines the remaining cumulative mining capabilities
502
Distributed ledger A determines the expected mining time based on the cumulative mining capabilities
503
Expected mining time too large?
NO
YES
504
Distributed ledger A switches to a new consensus mechanism
Distributed ledger A keeps original consensus mechanism
505
506
Destributed ledger A continues adding transactions to the distributed ledger

Fig. 5

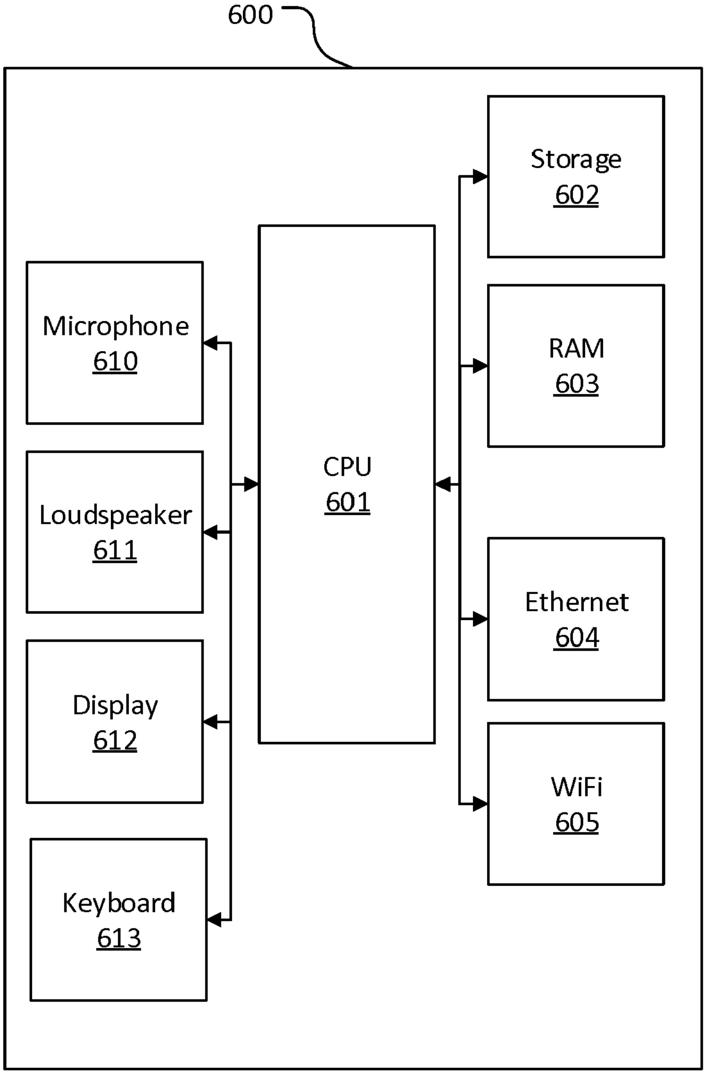
600
Storage
602
RAM
603
Microphone
610
CPU
601
Loudspeaker
611
Ethernet
604
Display
612
WiFi
605
Keyboard
613

Fig. 6

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 18 15 5717

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	US 2017/250815 A1 (CUENDE LUIS IVÁN [ES] ET AL) 31 August 2017 (2017-08-31) * paragraph [0004] - paragraph [0005]; figures 1-5 * * abstract * -----	1-6, 10-15,19	INV. H04L9/32 G06Q20/06
X	US 4 853 843 A (ECKLUND DENISE J [US]) 1 August 1989 (1989-08-01) * column 1 - column 3 * * column 31, line 32 - column 38, line 23 * * column 49, line 46 - column 50, line 40 * * figures 1-18 * -----	1-6, 10-15,19	
X	CN 105 488 675 A (BUBI NETWORK TECH CO LTD) 13 April 2016 (2016-04-13) * paragraph [0001] - paragraph [0034]; figure 1 * * abstract * -----	1-6, 10-15,19	

TECHNICAL FIELDS SEARCHED (IPC)
H04L G06Q G06F

~~The present search report has been drawn up for all claims~~

Place of search	Date of completion of the search	Examiner
Munich	22 June 2018	Hou, Jie

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
&amp; : member of the same patent family, corresponding document

Europäisches Patentamt
European Patent Office
Office européen des brevets

Application Number

EP 18 15 5717

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing claims for which payment was due.

□ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for those claims for which no payment was due and for those claims for which claims fees have been paid, namely claim(s):

□ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for those claims for which no payment was due.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

□ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

□ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

□ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☒ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

1-6, 10-15, 19

□ The present supplementary European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims (Rule 164 (1) EPC).

Europäisches Patentamt
European Patent Office
Office européen des brevets

LACK OF UNITY OF INVENTION SHEET B

Application Number

EP 18 15 5717

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. claims: 1-6, 10-15, 19

   Method, system and device for determining a common history between two distributed ledgers.

   ---

2. claims: 7-9, 16-18, 20

   Method, system and device for adopting a consensus algorithm of a distributed ledger.

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 18 15 5717

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-06-2018

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017250815 A1	31-08-2017	NONE	
US 4853843 A	01-08-1989	NONE	
CN 105488675 A	13-04-2016	NONE	

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82